



DESENVOLVIMENTO DE UMA FERRAMENTA COMPUTACIONAL EM PLATAFORMA ABERTA APLICADA À PREVISÃO DE VAZÕES AFLUENTES DAS USINAS HIDROELÉTRICAS

Fábio G. Ferreira

Anderson P. de Aragão

Patrícia T. L. Asano

Ricardo C. dos Santos

fabio.godoy@aluno.ufabc.edu.br

andersonparagao@gmail.com

patricia.leite@ufabc.edu.br

ricardo.caneloi@ufabc.edu.br

Curso de Engenharia de Energia, Universidade Federal do ABC

Avenida dos Estados, 5001, 09210-580, São Paulo, Santo André, Brazil

Rodrigo Sacchi

rodrigo.sacchi@ccee.org.br

Câmara de Comercialização de Energia Elétrica

Avenida Paulista, 2064 - 13º Andar, 01310-200, São Paulo, São Paulo, Brazil

Resumo. A eletricidade no Brasil é predominantemente gerada através de usinas hidroelétricas e termoelétricas, por isso o sistema é denominado sistema de geração hidrotérmico. Neste tipo de sistema, um dos objetivos do planejamento da operação é definir a quantidade de energia a ser gerada por cada tipo de usina de modo que o custo total da operação seja o mínimo possível dado um período de planejamento. Este é executado com o auxílio de modelos computacionais que realizam a previsão e geração de cenários estocásticos com base em modelos auto-regressivos. Uma alternativa à modelagem atual é a aplicação de técnicas de inteligência

computacional, as quais permitem a identificação de padrões e a aproximação de funções lineares e não-lineares das correlações temporais do processo estocástico das vazões afluentes. Logo, desenvolveu-se uma ferramenta amigável ao usuário baseada em redes neurais artificiais, e que foi aplicada ao problema de previsão de vazões, que é atividade essencial para determinar a geração hidroelétrica. Através dessa ferramenta, o usuário pode definir alguns dos parâmetros da rede neural artificial e avaliar os impactos de cada parâmetro nos resultados. A ferramenta foi totalmente desenvolvida em ambientes open source, que não exigem licença comercial podendo ser aplicada a problemas de previsão de séries temporais.

Keywords: *Previsão de vazões, Redes neurais artificiais, Planejamento Energético*

1 INTRODUÇÃO

Até Junho de 2016, as grandes usinas hidroelétricas representavam 61% da capacidade instalada brasileira, enquanto que os empreendimentos termoeletrônicos representavam 28%. Esses números demonstram que, no Brasil, a maior parte da eletricidade é gerada a partir das fontes de energia hidráulica e térmica, tanto que o sistema elétrico brasileiro é classificado como um sistema hidrotérmico. As justificativas para a predominância dessas fontes no Brasil são diversas, dentre elas vale citar o alto potencial hidráulico do país, o relevo brasileiro, a possibilidade de armazenar água em reservatórios, e o fato de que as termoeletrônicas podem ser acionadas por questões de segurança durante períodos de seca, por exemplo.

No entanto, uma desvantagem de um sistema como o brasileiro é que a atividade de planejamento da operação do mesmo é bastante complexa, uma vez que se faz necessário definir um equilíbrio entre geração térmica, que é mais cara devido ao impacto do preço dos combustíveis, e a geração hidroelétrica, que é mais barata porque utiliza água como "combustível". Então, o operador do sistema precisa fazer a decisão de deplecionar (usar água) ou não deplecionar (usar térmica) os reservatórios, sendo que as consequências de tal decisão devem ser analisadas como mostra a Fig. 1 (Silva, 2001).

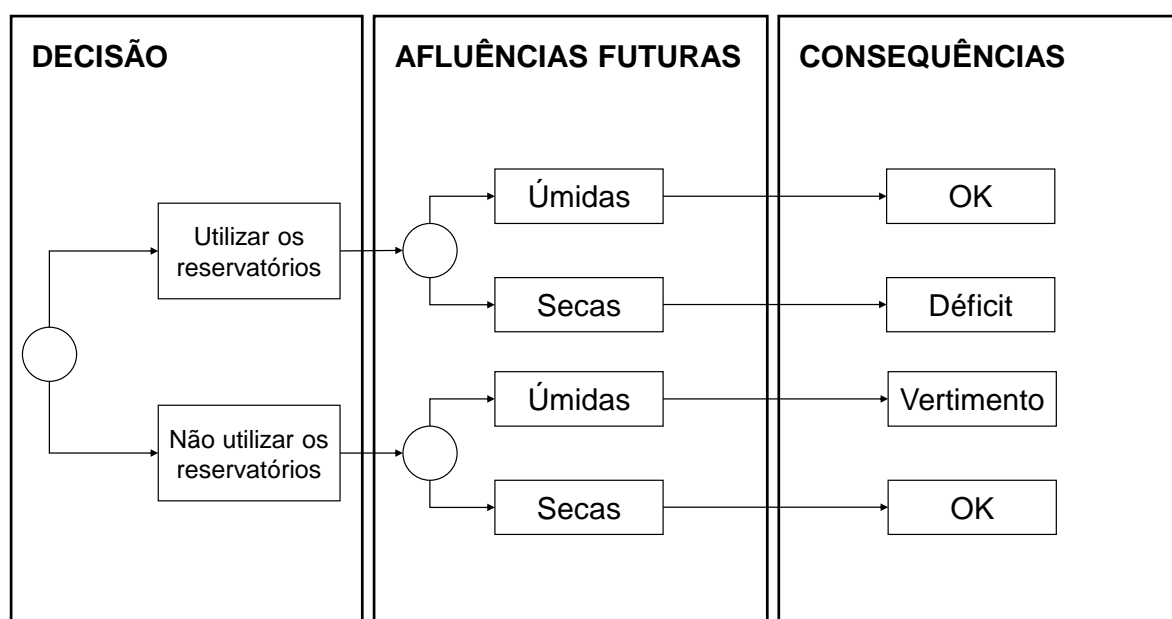


Figura 1: Decisão - Sistemas Hidrotérmicos

Desse modo, o operador deve analisar diferentes cenários hidrológicos para determinar a quantidade de recurso que será utilizada visando baixos custos operacionais e o aumento da segurança no fornecimento de energia elétrica para o país.

Para resolver esse problema, é importante perceber que as fluências futuras são de natureza estocástica, isto é, não é possível modelá-las com modelos determinísticos, e isso implica na utilização de modelos de análise de séries temporais em conjunto com técnicas de programação linear para que se possa fazer simulações com diferentes cenários hidrológicos.

Atualmente para a realização e tomada de decisão da programação da operação do sistema hidrotérmico, o ONS faz uso de dois modelos computacionais desenvolvidos pelo Centro de Pesquisas de Energia Elétrica (CEPEL - RJ), são eles: **NEWAVE** e **DECOMP**, que são usados para determinar a operação do sistema; e de quatro modelos hidrológicos o **PREVIVAZ**, **PREVIVAZH**, **PREVIVAZM** e **GEVAZP**, que servem de suporte para geração de cenários hidrológicos e previsão das afluições futuras. A seguir é feita uma breve descrição de cada modelo (Maceira et al., 2013b,a; Deus, 2008):

- **NEWAVE** - ferramenta de estudo de médio prazo, que é utilizada para definir o planejamento num horizonte de 5 anos por etapas mensais considerando a representação a sistemas equivalentes. O principal resultado deste modelo é a função de custo futuro;
- **DECOMP** - ferramenta de curto prazo, que é utilizada para definir o planejamento num horizonte que vai de 2 a 6 meses por etapas semanais considerando a representação individual das usinas hidroelétricas;
- **PREVIVAZ** - é um modelo utilizado para realizar a previsão das afluições abrangendo um período que vai de uma a até seis semanas adiante. Este utiliza os seguintes modelos estatísticos: autoregressivos $AR(p)$, autoregressivos e de médias móveis $ARIMA(p,q)$ ou o periódico $PARMA(p,q)$ até quarta ordem;
- **PREVIVAZH** - em algumas situações as previsões realizadas pelo **PREVIVAZ** podem estar incompletas, portanto o **PREVIVAZH** é utilizado de forma complementar ao **PREVIVAZ** gerando previsões de vazões diárias num horizonte que vai de 7 a 13 dias;
- **PREVIVAZM** - é uma ferramenta especial que é utilizada para estudos de verificação de condições de atendimento da demanda energética no horizonte anual;
- **GEVAZP** - este modelo gera um conjunto de cenários sintéticos de vazões e energias utilizando o modelo $PAR(p)$, que modela a afluição de um mês como uma combinação linear das afluições dos meses anteriores e de uma componente aleatória. Para o modelo **DECOMP**, o **GEVAZP** é utilizado para gerar os cenários de vazões a partir do segundo mês, enquanto que para **NEWAVE**, o **GEVAZP** gera cenários de energia afluente para até cinco anos a frente com base no valor de afluição do mês anterior ao primeiro mês do planejamento;

Em (Guilhon e Rocha, 2007) foi publicado um estudo avaliando a precisão dos modelos **PREVIVAZ** e **PREVIVAZH**. Alguns dos resultados baseados na métrica de *Mean Absolute Percentage Error* (MAPE) podem ser vistos na Tab. 1.

Tabela 1: MAPE dos modelos Previvaz (1997 a 2004) e Previvazh (2000 a 2001)

Usina	Desvio Previvaz (%)	Desvio Previvazh (%)
Itaipu	15,5	30,4
Furnas	27,9	24,2
Paraibuna	28,6	37,2
Três Marias	35,3	31,8
Serra da Mesa	35,4	51,7
Passo Real	42,0	27,9
Itá	67,7	-

É possível observar que para algumas das usinas os desvios são consideráveis, indicando que é importante a aplicação de outros métodos que permitam realizar melhores previsões. Dentre as metodologias modernas existentes, destaca-se a técnica de inteligência artificial, denominada Rede Neural Artificial (RNA). Estas foram concebidas com base no sistema nervoso humano, e são implementadas de maneira a "imitar" as características de plasticidade e renovação do cérebro humano que são a base do aprendizado por experiência (Haykin, 2009).

Estruturalmente, as RNAs possuem como unidades mais básicas os neurônios, que são agrupados em três tipos de camadas: entrada, intermediária(s) e saída. Os neurônios da camada de entrada devem ser conectados aos neurônios da camada intermediária que por sua vez devem estar conectados aos neurônios da camada de saída. A quantidade de neurônios em cada camada, assim como o próprio número de camadas são dois dos parâmetros variáveis de uma RNA, e, geralmente, esses valores são variados até encontrar-se aqueles que gerem os melhores resultados de acordo com métricas que podem ser escolhidas pelo usuário do modelo de RNA (Haykin, 2009).

As RNAs são classificadas com base nas conexões entre os neurônios das diferentes camadas e no número de camadas que constituem a rede. Todavia, há algumas classificações que são consideradas como principais: redes neurais artificiais *feedforward* de camada simples e de camadas múltiplas, redes recorrentes e redes reticuladas. A terminologia *feedforward* é utilizada para indicar que o fluxo de informações é unidirecional, ou seja, vai da camada de entrada até a camada de saída (Silva et al., 2010). A Figura 2 a seguir ilustra a rede *feedforward* de camadas múltiplas:

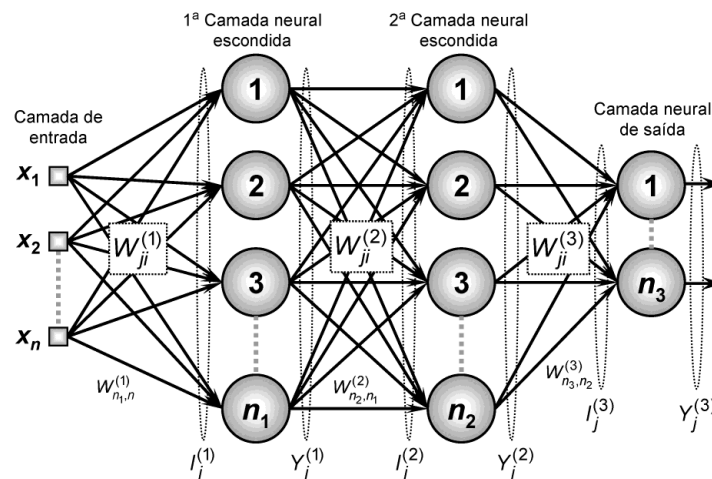


Figura 2: Rede *feedforward* de camadas múltiplas (Silva et al., 2010)

As conexões entre os neurônios da RNA da Fig. 2 são denominados de pesos sinápticos e representados por W_{ji}^L , sendo que L representa o número da camada, e o índice ji indica que W^L é o peso sináptico conectando o j -ésimo neurônio da camada L ao i -ésimo neurônio da camada $L-1$ (Silva et al., 2010). Esses pesos possuem valores que sofrem atualizações ao longo do processo de aprendizado da RNA, ou seja, com o decorrer do tempo de execução do modelo, os pesos sinápticos vão sofrendo modificações até que haja convergência a uma resposta final, que é a saída da rede. Em outras palavras, os pesos sinápticos armazenam as informações sobre o problema estudado, e o algoritmo de treinamento atualiza tais pesos ao longo do aprendizado.

O processo de aprendizado citado anteriormente pode ser classificado de duas maneiras: supervisionado e não-supervisionado. No treinamento supervisionado, é apresentado um par de dados ao modelo, sendo que um dos dados é um sinal de entrada, e o outro é a saída do modelo para aquele dado da entrada. A partir daí, o dado de entrada é apresentado à RNA, que fará o treinamento e ajuste dos pesos sinápticos até que o processo gere um dado como resultado (um dado de saída que é baseado no dado de entrada), esse dado de saída será comparado ao dado de saída real, caso a diferença não seja superior a uma certa tolerância, pode-se dizer que o resultado produzido pela rede foi satisfatório, do contrário é preciso repetir o treinamento. Já no treinamento não-supervisionado, não existe um par de dados, apenas a entrada é conhecida, e a partir daí a rede neural precisa fazer os ajustes necessários nos parâmetros para determinar um modelo com boa generalização (Abu-Mostafa et al., 2012).

Adicionalmente, o aprendizado pode ser realizado de duas maneiras diferentes: por lote ou *online*. Neste último, N pares entrada-saída são apresentados à rede e os pesos são atualizados N vezes, enquanto que no primeiro todos os N pares entrada-saída são apresentados, mas os pesos são atualizados apenas uma vez (Haykin, 2009).

Os trabalhos de (Ballini, 2000), (Faria, 2009), (Fayal, 2008), (Lachtermacher e Fuller, 1994) estudaram a aplicação de RNAs ao problema de previsão de vazões. A diferença deste artigo, com relação aos demais, está na inovação do desenvolvimento da ferramenta computacional que foi implementada utilizando-se dos conceitos de redes neurais artificiais, mas em uma plataforma aberta, ao invés de ferramentas ou pacotes prontos.

2 METODOLOGIA

2.1 Algoritmo de retropropagação

O processo de aprendizado citado na seção anterior é realizado através de um dado algoritmo. Na realidade, existem diversos algoritmos de aprendizado, e cada um possui suas vantagens e desvantagens. Neste trabalho, a ferramenta computacional aplica o algoritmo de retropropagação (*backpropagation*), que é baseado no método do gradiente descendente (Baliní, 2000). Este algoritmo altera os pesos da rede de modo a minimizar a diferença entre as saídas desejadas e as saídas fornecidas pelo modelo, sendo que tal diferença é medida pela função erro quadrático mostrada a seguir:

$$E = \frac{1}{2} \sum_j (d_j - y_j)^2 \quad (1)$$

Nesta equação, j é o índice referente à saída do j -ésimo neurônio da camada de saída, e y_j e d_j são o valor de saída dado pelo modelo e o valor de saída desejado, respectivamente.

A minimização de E pelo gradiente descendente ocorre em duas etapas denominadas *forward* e *backward* (Rumelhart et al., 1986). Utilizando as mesmas notações da Fig. 2, a etapa *forward* é descrita por duas simples equações: a primeira representa a entrada ponderada I_j^L ao j -ésimo neurônio de uma camada L :

$$I_j^L = \sum_i x_i w_{ji}^L \quad (2)$$

E a outra é a equação de uma função de ativação não-linear como a logística, que é aplicada à entrada ponderada I_j^L , e representa a saída do j -ésimo neurônio da camada L :

$$y_j^L = \frac{1}{1 + e^{-x_i w_{ji}^L}} \quad (3)$$

Como citado anteriormente, o índice ji nas Eq. (2) e (3) significa que aquele é o peso sináptico w^L conectando o j -ésimo neurônio da camada L ao i -ésimo neurônio da camada $L-1$ ou ao i -ésimo sinal de entrada caso L seja a camada de entrada.

Já a etapa *backward* é mais complicada, porque é preciso propagar as derivadas da camada de saída até a camada de entrada indicando que quanto mais camadas, maior a complexidade desta fase. Assim sendo, a explicação da fase *backward* apresentada neste artigo será baseada em uma rede que possui apenas uma camada intermediária, que será identificada por $L-1$ e uma camada de saída, que será identificada por L .

O primeiro passo, é realizar o cálculo da derivada parcial $\partial E / \partial y$ para os neurônios da camada de saída. Logo, a partir da Eq. (1), chega-se a:

$$\frac{\partial E}{\partial y_j} = d_j - y_j \quad (4)$$

Em seguida, é preciso determinar a variação dos sinais de entrada aos neurônios da camada de saída com relação à variação do peso sináptico, o que pode ser feito através do cálculo da derivada parcial $\partial I_j^L / \partial w_{ji}^L$

$$\frac{\partial I_j^L}{\partial w_{ji}^L} = y_j^{L-1} \quad (5)$$

A Eq. (5) possui tal resultado, porque os valores de entrada aos neurônios da camada de saída são os valores de saída dos neurônios da camada intermediária.

Por fim, calcula-se a variação do sinal de saída dos neurônios da camada de saída com relação ao sinal de entrada aos mesmos. Novamente, isso é possível através do cálculo de derivadas parciais, que neste caso é o resultado da derivada da função logística (Eq. 3).

$$\frac{\partial y_j^L}{\partial I_j^L} = y_j^L(1 - y_j^L) \quad (6)$$

Juntando os resultados das Eqs. (4), (5) e (6), é possível calcular a variação da função erro total com relação ao ajuste dos pesos sinápticos

$$\frac{\partial E}{\partial w_{ji}^L} = (d_j - y_j)[y_j^L(1 - y_j^L)]y_j^{L-1} \quad (7)$$

E esta última equação é a que pode ser utilizada para atualizar os pesos sinápticos da camada de saída.

Após calcular o ajuste dos pesos sinápticos da camada de saída, é preciso dar sequência a fase *backward* e calcular qual deve ser o ajuste dos pesos sinápticos da camada intermediária de maneira similar ao que foi feito para a camada de saída com a diferença de que é mais fácil calcular a variação dos sinais de entrada aos neurônios da camada intermediária com relação ao ajuste dos pesos sinápticos. Portanto:

$$\frac{\partial I_j^L}{\partial w_{ji}^L} = x_i \quad (8)$$

Isto porque o termo I_j^L vem da Eq. (2).

A Eq. (8) pode ser utilizada para calcular a variação da saída dos neurônios da camada intermediária com relação ao sinal de entrada dos mesmos que, neste caso, é o sinal de entrada da rede neural, enquanto que para os neurônios da camada de saída o sinal de entrada era determinado pela saída dos neurônios da camada intermediária. Logo:

$$\frac{\partial y_j^L}{\partial I_j^L} = y_j^L(1 - y_j^L) \quad (9)$$

que é igual à Eq. (6), porque a função logística foi utilizada novamente.

Finalmente, é preciso calcular o termo $\partial E / \partial y_j^L j$ que pode ser obtido através dos seguintes passos:

$$\frac{\partial E}{\partial y_j^L} = \sum_{k=1}^{n_2} \frac{\partial E}{\partial I_k^{L+1}} \cdot \frac{\partial I_k^{L+1}}{\partial y_j^L} \quad (10)$$

$$\frac{\partial E}{\partial y_j^L} = \sum_{k=1}^{n_2} \frac{\partial E}{\partial I_k^{L+1}} \cdot w_{kj}^{L+1} \quad (11)$$

Sendo que é importante observar que na Eq. (11) o índice do termo do peso sináptico é kj o que significa que aquele é o peso sináptico conectando os neurônios da camada de saída a um neurônio j da camada intermediária.

Agora basta juntar os termos das Eqs. (8), (9) e (11)

$$\frac{\partial E}{\partial w_{ji}^L} = \sum_{k=1}^{n_2} \frac{\partial E}{\partial I_k^{L+1}} \cdot w_{kj}^{L+1} \cdot y_j^L (1 - y_j^L) \cdot x_i \quad (12)$$

Ambas as Eqs. (7) e (12) são utilizadas na equação abaixo, que é a equação do gradiente descendente.

$$\Delta w = -\eta \frac{\partial E}{\partial w} \quad (13)$$

Onde η é denominado taxa de aprendizagem, e o sinal negativo representa a busca por uma mudança no termo w_{ji} que reduza o valor de E . Um alto valor de η resulta em aprendizagem rápida, porém também aumenta o risco da rede tornar-se instável provocando a saturação dos neurônios. Se a taxa for baixa, as mudanças nos pesos sinápticos serão cada vez menores, logo aumenta a espera até que a condição de convergência seja atingida (Ballini, 2000).

No entanto, uma simples modificação na Regra Delta (Eq. 13), que permite aumentar a velocidade de aprendizagem, é o uso de um termo denominado *momentum*, α , que varia de 0 a 1, e determina a contribuição relativa do gradiente atual e os gradientes anteriores a atualização do peso (Ballini, 2000; Rumelhart et al., 1986). Com esta alteração, a Eq. (13) passa a ser:

$$\Delta w(t) = -\eta \frac{\partial E}{\partial w(t)} + \alpha \Delta w(t-1) \quad (14)$$

que é conhecida como Regra Delta Generalizada (Ballini, 2000; Rumelhart et al., 1986).

2.2 Desenvolvimento do algoritmo

O algoritmo descrito acima foi implementado utilizando a linguagem de programação Java, utilizando a biblioteca gráfica JavaFX para criação da interface com o usuário. Os dados de vazão de todos os postos de medição foram armazenados em um banco de dados criado por meio do Sistema Gerenciador de Banco de Dados *MySQL*. Todas essas ferramentas são *softwares* livres e amplamente utilizados. A seguir estão discutidos os dados utilizados no trabalho assim como arquitetura da RNA escolhida, entre outros.

Série histórica. Uma série histórica de vazões afluentes é uma sequência de valores mensais correspondente à média dos valores instantâneos observados durante o mês, em um posto fluviométrico (Silva et al., 2010). A série histórica de vazões para os aproveitamentos hidrelétricos que fazem parte do Sistema Interligado Nacional (SIN) foram obtidas através do arquivo VAZOES.DAT, que foi extraído do *deck* de preços do sistema NEWAVE, e que possui dados para 212 postos de medição de vazões. A leitura do arquivo VAZOES.DAT foi realizada com o auxílio da ferramenta VazEdit desenvolvida pelo ONS, e disponível para download no sítio da Câmara de Comercialização de Energia Elétrica (CCEE). Depois de extraído do *deck*, o arquivo gerado pelo VazEdit foi convertido para *comma separated value* (CSV), pois o *MySQL* possui uma ferramenta para importação de dados contidos em arquivos CSV o que facilitou o trabalho.

Em (Sacchi, 2009), para o treinamento da RNA utilizaram-se os dados entre o período de 1933 até 1994 excetuando-se as vazões do período 1972 a 1976, que apresentam um comportamento hidrológico médio e que por isso foram utilizados na fase de teste dos parâmetros ajustados durante o treinamento. Neste projeto, optou-se por utilizar o mesmo conjunto de dados, sendo que 75% dos valores do conjunto de treinamento foram utilizados para ajuste

dos parâmetros, e os 25% restantes foram utilizados para validação do modelo seguindo as recomendações de (Haykin, 2009).

Uma vez que os dados foram adicionados ao banco de dados, por meio de instruções *SQL* extraiu-se apenas os dados de vazão contidos nos intervalos desejados, assim como pôde-se determinar a vazão máxima para qualquer um dos postos contidos no arquivo VAZOES.DAT.

Dentre as séries históricas dos 212 postos, escolheu-se focar o estudo no posto de Furnas, que faz parte da bacia do Rio Paraná, devido a sua importância e localização (Ballini, 2000; Sacchi, 2009).

Pré-processamento. Apesar da variada gama de técnicas de pré-processamento existentes, neste trabalho fez-se uso apenas do procedimento de normalização dos dados, isto é, faz-se a divisão de todos os valores de vazões da série histórica pela máxima vazão registrada para a usina. Desse modo, ficou garantido que os valores de entrada da rede estariam no intervalo [0,1], garantindo a estabilidade numérica da RNA (Sacchi, 2009), assim como a compatibilidade com a função sigmóide (ou logística) já que o domínio de tal função é [0,1].

Além disso, foi necessário manipular o vetor que armazenava os dados extraídos do banco de dados. Neste caso, os dados foram manipulados para que eles ficassem de acordo com o que é definido pela teoria das redes perceptron multicamadas de entradas atrasadas no tempo com configuração focada (do inglês, *focused time-lagged feedforward network* - TDNN). Esta é utilizada para prever valores futuros a partir de um instante qualquer t em função do conhecimento dos valores temporariamente anteriores. Matematicamente:

$$x(t) = f(x(t-1), x(t-2), \dots, x(t-n_p)) \quad (15)$$

onde n_p é a ordem do preditor que determina a quantidade de amostras passadas necessárias para realizar a previsão, que é o mesmo que determinar o número de neurônios da camada de entrada. É importante notar que a Eq. (15) é da família dos modelos auto-regressivos (Silva et al., 2010).

As redes perceptron multicamadas de entradas atrasadas no tempo com configuração focada recebem este nome, porque uma linha de atraso é inserida apenas à camada de entrada da rede, sendo que, quando foi concebida a ideia, os atrasos eram inseridos em todas as camadas. A linha de atraso atua como uma memória, isto é, as amostras anteriores são sempre inseridas dentro da rede sem que haja necessidade de realimentação. A Figura 3 abaixo, ilustra a topologia da rede (Silva et al., 2010).

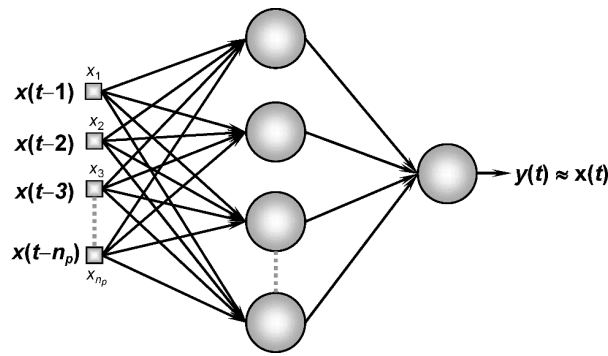


Figura 3: Rede com atraso de tempo focado

De acordo com a Fig. 3 acima, a rede tentará ajustar as matrizes de pesos visando minimizar o erro $E(t)$, que é a diferença entre $x(t)$ e $y(t)$.

Para o caso deste estudo optou-se por utilizar a informação das vazões de 12 meses como entrada, para então fazer a previsão para um mês seguinte qualquer, isto é, poder-se-ia estimar a vazão de qualquer mês utilizando as vazões dos 12 meses anteriores àquele mês.

A Figura 4 exemplifica como ocorreu a montagem dos vetores que foram utilizados para o treinamento e teste da rede.

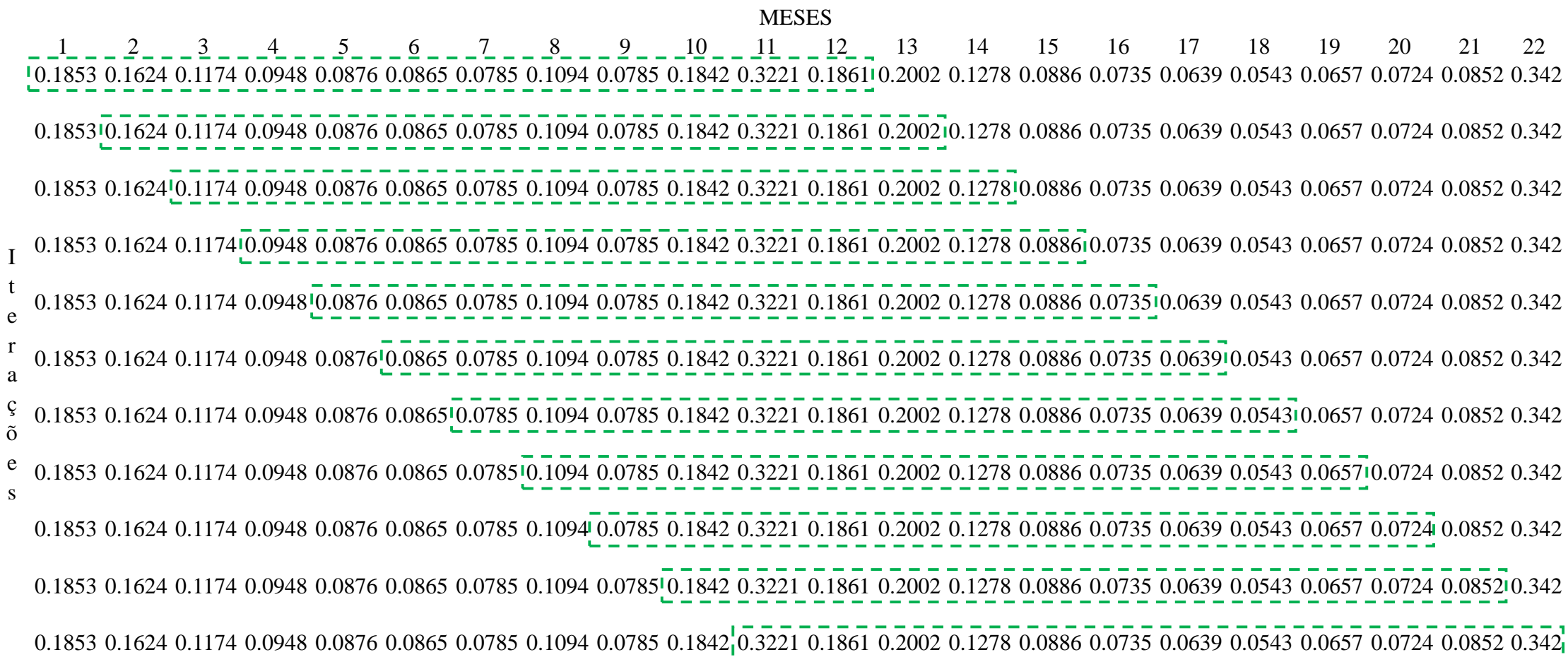


Figura 4: Janela deslizante ilustrando o processo de montagem dos vetores (Adaptado de Silva et al. (2010))

As linhas tracejadas da Fig. 4 representam uma "janela" em que o último valor mais a direita da janela é aquele que será aproximado utilizando os onze valores a esquerda. Em seguida, a "janela" é movida para a direita de modo que o valor que fora previsto na etapa anterior será utilizado para calcular o próximo valor e assim por diante até que seja feita o número de previsões desejada.

Definição dos parâmetros da RNA. Para o sucesso do algoritmo de RNA baseado no método de retropropagação é importante a definição dos seguintes parâmetros: número de camadas, número de neurônios nas camadas de entrada, intermediária e de saída, inicialização dos pesos sinápticos, taxa de aprendizagem, termo momento, ordem do termo preditor e função de ativação (Silva et al., 2010).

Devido ao grande número de parâmetros livres, e também ao grande número de combinações possíveis, diversos autores definiram heurísticas que permitem estimar tais parâmetros com maior precisão mais rapidamente. No entanto, neste trabalho optou-se pelo processo de tentativa e erro que, apesar de muito mais trabalhoso, permitiu entender melhor o processo de aprendizado do algoritmo. Logo, foram feitas diversas observações do comportamento do algoritmo até atingir a configuração que obteve melhor resultado.

Começando pelo número de camadas da rede neural, ficou definido que ela possuiria apenas três camadas (uma de entrada, uma intermediária, e uma de saída), porque a mesma arquitetura já fora utilizada para estudar o mesmo problema (Ballini, 2000; Sacchi, 2009). Para determinar o número de neurônios da camada de entrada, definiu-se que tal número e a ordem do termo preditor seriam iguais, e que o número de neurônios da camada intermediária seria igual a no máximo duas vezes mais um o número de neurônios da camada de entrada, e haveria a presença de apenas um neurônio na camada de saída seguindo a metodologia das TDNN.

A inicialização dos pesos sinápticos seguiu a regra aleatória, isto é, como não se tinha nenhuma informação *a priori* que permitisse supor um bom valor para a inicialização destes parâmetros, permitiu-se que os pesos fossem determinados de maneira aleatória pelo próprio algoritmo (Haykin, 2009), e que eles seriam atualizados pelo algoritmo de retropropagação até que a diferença entre o erro quadrático médio de duas iterações consecutivas fosse menor do que 0.5×10^{-6} .

Dos parâmetros restantes (com exceção da função de ativação que já teve sua escolha justificada), a sua definição ocorreu ao longo da fase de teste do algoritmo, ou seja, após finalizar o algoritmo, foi necessário variar tais parâmetros e avaliar a resposta final até que um resultado satisfatório fosse atingido.

Por fim, após toda a fase de tentativa e erro, concluiu-se que as redes com melhores resultados foram aquelas que possuíam: 12 neurônios na camada de entrada (igual a ordem do termo preditor), 25 neurônios na camada intermediária, taxa de aprendizagem igual a 0.26 e termo *momentum* igual a 0.1; 12 neurônios na camada de entrada, 24 neurônios na camada intermediária, taxa de aprendizagem igual a 0.2 e termo *momentum* igual a 0.03; 12 neurônios na camada de entrada, 24 neurônios na camada intermediária, taxa de aprendizagem igual a 0.2 e termo *momentum* igual a 0.05.

3 RESULTADOS

Nesta seção será apresentada a interface gráfica da ferramenta, bem como a análise dos resultados alcançados.

Observe que nas Figs. 5 e 6 são apresentadas as telas finais da ferramenta, que tem como objetivo tornar a ferramenta bastante amigável aos usuários e também possibilita a utilização da mesma em treinamentos e ensino.



Figura 5: Tela inicial do sistema

Para que se possa utilizar a ferramenta apresentada na Fig. 5 se faz necessário escolher um dos 212 postos hidrológicos disponíveis pelo ONS. Assim que a escolha é feita, o gráfico da Fig. 6 é gerado para que o usuário possa ver a série histórica completa de vazões.

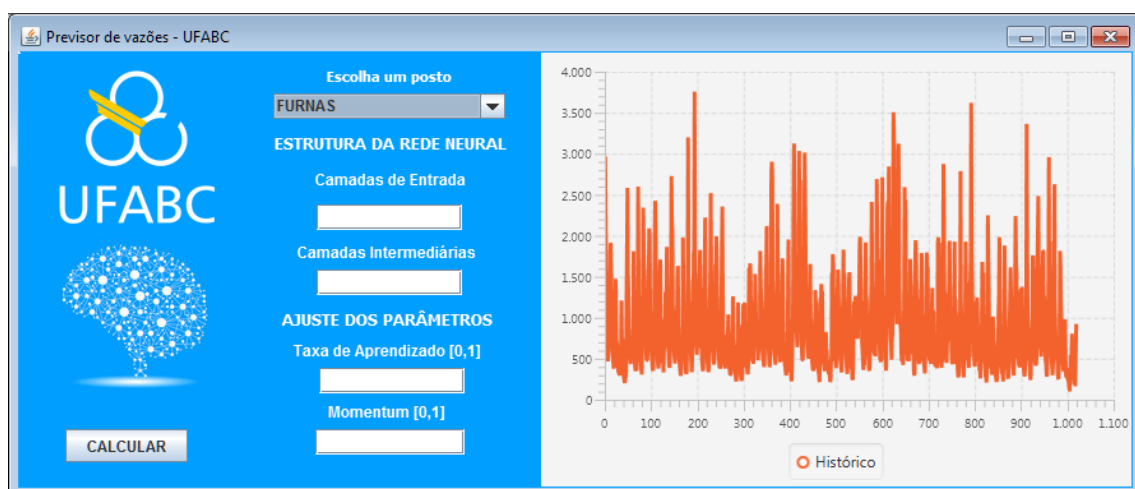


Figura 6: Tela após escolha do posto a analisar

Após escolhido o posto, o usuário pode selecionar as seguintes características da rede: número de neurônios na camada de entrada, número de neurônios na camada intermediária, taxa de aprendizado e termo *momentum*.

Como fora explicado em seções anteriores o número de camadas foi fixado em três, assim como a quantidade de neurônios na camada de saída foi fixada em um. Do mesmo modo, a ferramenta foi testada utilizando 12 neurônios na camada de entrada, 25 e 24 neurônios na camada intermediária, taxa de aprendizagem igual a 0.2 e 0.26 e termo *momentum* igual a 0.03, 0.05 e 0.1. A escolha do número de neurônios nas camadas de entrada e intermediária foi feita de modo a evitar o *overfitting*, que é quando a RNA "memoriza" as respostas ao invés de "aprender" e ser capaz de gerar novos resultados. Já as taxas de aprendizado e *momentum* foram selecionados após a realização de diversas simulações, sendo interessante notar que baixos valores de taxa de aprendizado e termo *momentum* geraram os melhores resultados, que podem ser vistos nos gráficos a seguir (em todos os gráficos o eixo das ordenadas representa as vazões normalizadas, isto é, os valores são adimensionais):

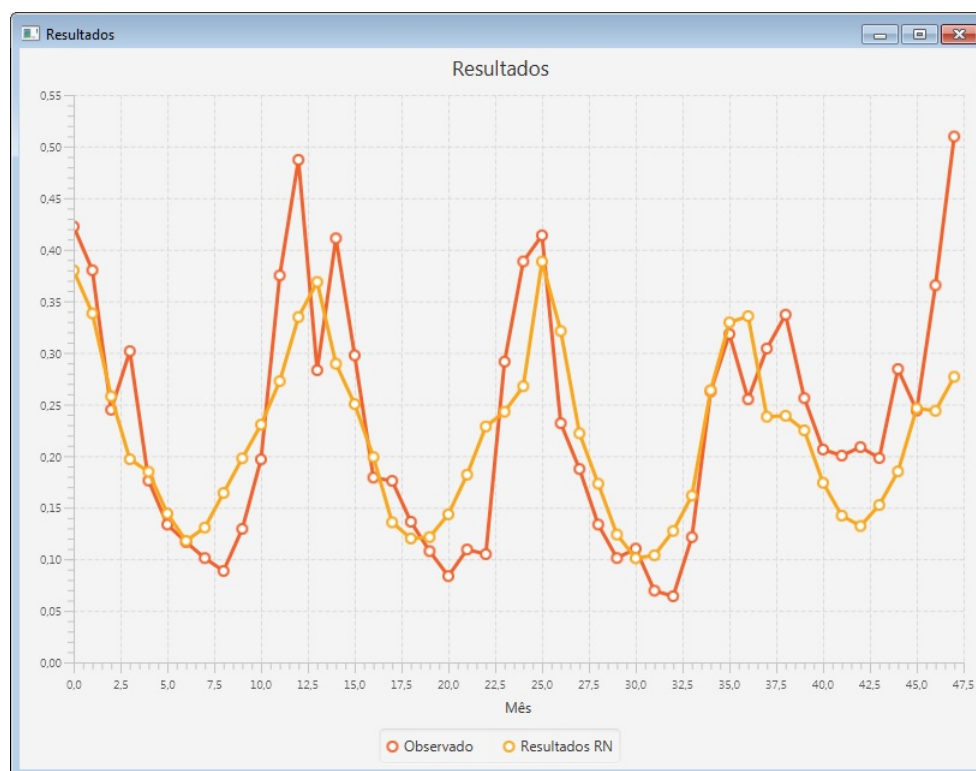


Figura 7: 12 neurônios na camada de entrada, 25 na camada intermediária, taxa de aprendizado 0.26 e termo *momentum* 0.1

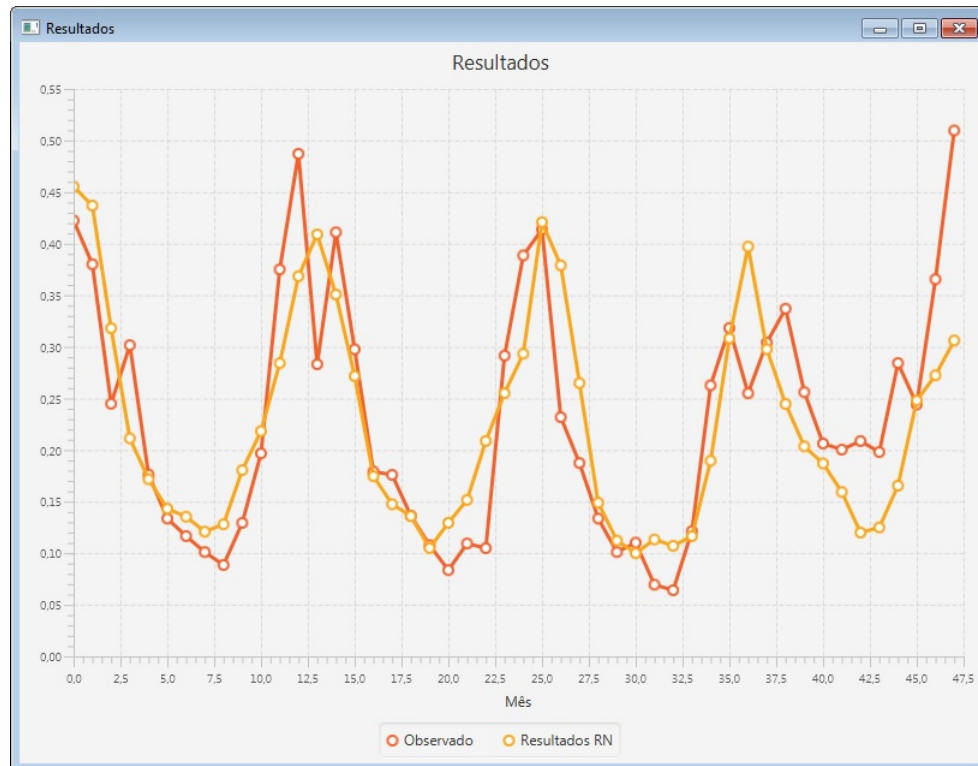


Figura 8: 12 neurônios na camada de entrada, 24 na camada intermediária, taxa de aprendizado 0.2 e termo *momentum* 0.03

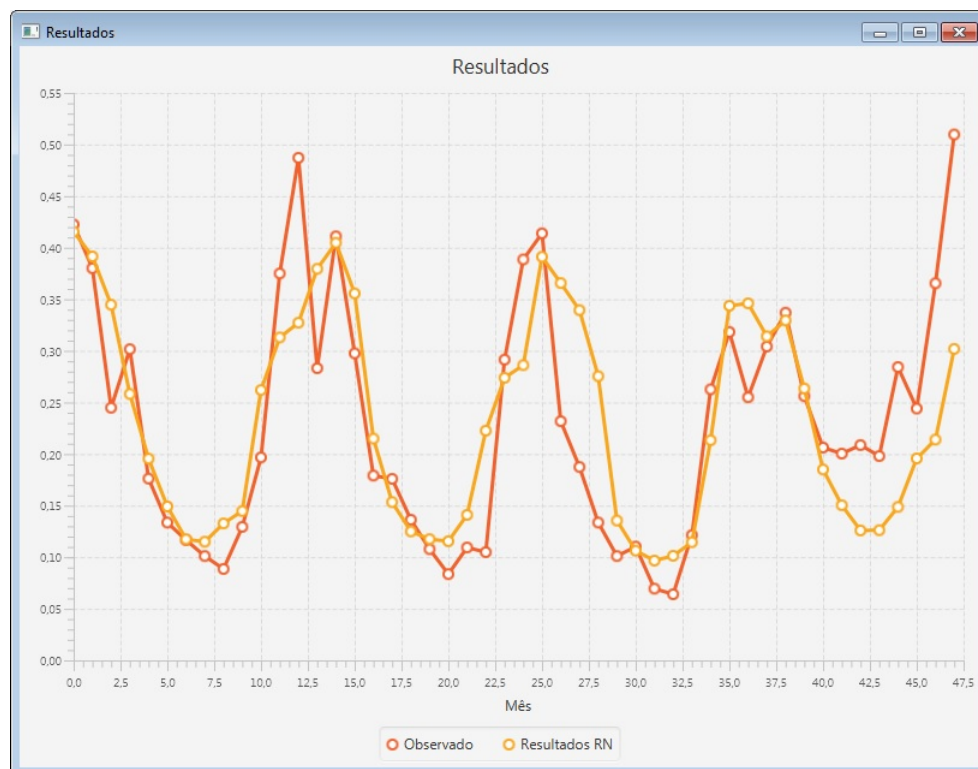


Figura 9: 12 neurônios na camada de entrada, 24 na camada intermediária, taxa de aprendizado 0.2 e termo *momentum* 0.05

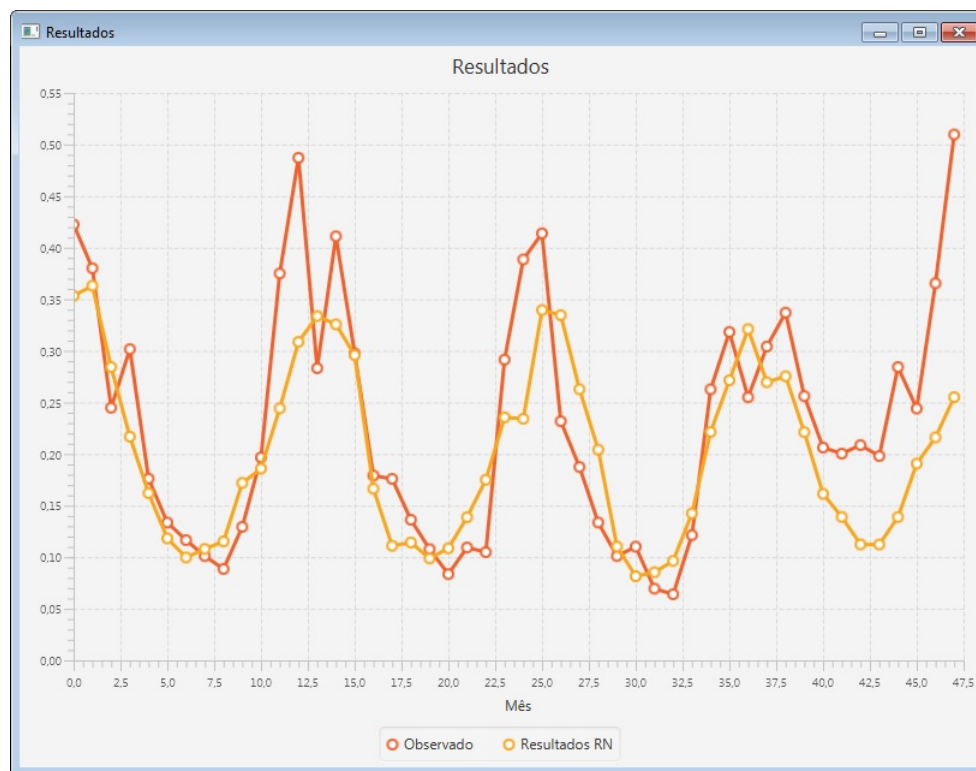


Figura 10: 12 neurônios na camada de entrada, 25 na camada intermediária, taxa de aprendizado 0.2 e termo *momentum* 0.03

Num primeiro momento a análise visual dos gráficos permitia avaliar o resultado da fase de teste, que foi realizada com um conjunto de dados diferente daqueles utilizados nas etapas de treinamento e validação. Isto é, caso os pontos das curvas *Observado* e *Resultados RN* fossem muito distantes ficava evidente que o objetivo, que era realizar a previsão das vazões afluentes, não foi atingido, e que o processo deveria ser repetido. Nas Figs. 7, 8, 9 e 10 é possível ver que as curvas acompanham a mesma tendência, e alguns pontos das curvas são coincidentes.

Em seguida à leitura dos gráficos, foi necessário analisar as métricas estatísticas adimensionais Erro Quadrático Médio (EQM), Erro Percentual Médio (EPM) e Erro Relativo Médio (MRE) para quantificar a qualidade dos resultados. O resultado das métricas EQM, EPM e MRE pode ser observado na Tab. 2 abaixo, assim como o número de épocas de cada um dos resultados:

Tabela 2: Métricas para avaliação dos resultados do conjunto de teste

Gráfico	Épocas	EQM	EPM (%)	MRE
Figura 7	92	5.43E-3	28.72	1.43E-1
Figura 8	152	5.10E-3	25.64	1.03E-1
Figura 9	117	5.59E-3	26.68	1.33E-1
Figura 10	93	6.08E-3	25.47	8.79E-2

Nos resultados acima, nota-se que, como não foram feitas mudanças significativas nos

parâmetros livres (número de neurônios e taxas de aprendizado e *momentum*) da RNA, o número de épocas não foi muito diferente para cada simulação. Do mesmo modo, as métricas da Tab. 2 também não resultaram em valores muito discrepantes uns dos outros, e que são comparáveis aos resultados expostos na Tab. 1, com destaque aos resultados das Figs. 8, 9 e 10, que foram ligeiramente superiores aos resultados obtidos utilizando o modelo PREVIVAZ.

4 CONCLUSÕES

A previsão das vazões afluentes é uma atividade de alta complexidade e que deve ser estudada através de diferentes técnicas de análise de série temporal para que seja possível identificar quais os modelos mais adequados. Neste trabalho foi escolhida a técnica das redes neurais artificiais, que já foram estudadas por outros autores, e que geraram resultados promissores.

Diferente dos demais autores, que na maioria dos casos utilizaram ferramentas já disponíveis, foi feito um estudo do algoritmo de retropropagação, que é um dos mais empregados às RNAs, para implementá-lo em linguagem de programação Java. Além da implementação do algoritmo, também foi dado foco à interface gráfica da ferramenta para que a mesma fosse amigável ao usuário. Para tanto a interface possui poucos campos para receber as entradas do usuário e faz uso de gráficos para exibir as séries históricas e os resultados.

Ao longo da realização da etapa de simulações encontrou-se dificuldade para melhorar os resultados gerados pela RNA. Em parte, isto é positivo já que indica boa capacidade de generalização da rede, mas também indica que para conseguir melhores resultados talvez fosse necessário utilizar outras metodologias na etapa de pré-processamento de dados ou até testar outros algoritmos de aprendizado diferentes do algoritmo de retropropagação, que podem possuir características específicas capazes de melhorar ainda mais a performance da previsão de vazões.

Ainda assim, ressalta-se novamente que foi possível obter resultados melhores do que aqueles produzidos pelo modelo PREVIVAZ, e também melhor do que os resultados obtidos pelos modelos SONARX e SONARX-RBF, sem pré-processamento dos dados, obtidos em (Sacchi, 2009) para uma série similar a utilizada neste trabalho. O que demonstra ser promissora a investigação da aplicação das RNA na tarefa de previsão de vazões afluentes às usinas hidroelétricas.

Agradecimentos

A Universidade Federal do ABC que permitiu a realização deste trabalho.

REFERÊNCIAS

- Abu-Mostafa, Y., Magdon-Ismail, M., e Lin, H., 2012. *Learning from Data: A Short Course*. AMLBook.com.
- Ballini, R., 2000. *Análise e Previsão de Vazões Utilizando Modelos de Séries Temporais, Redes Neurais e Redes Neurais Nebulosas*. Tese de Doutorado, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

- Deus, M. L. D., 2008. *Séries temporais aplicadas ao Planejamento da Operação do Sistema Interligado Nacional*. Tese de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro.
- Faria, D. S., 2009. *Modelagem e Previsão de Vazões de Rios baseadas em Climatologia*. Tese de Mestrado, UFRJ/COPPE.
- Fayal, M. A. A., 2008. *Previsão de Vazão por Redes Neurais Artificiais e Transformada Wavelet*. Tese de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro.
- Guilhon, L. G. F. e Rocha, V. F., 2007. Comparação de métodos de previsão de vazões naturais afluentes a aproveitamentos hidroelétricos. *Revista Brasileira de Recursos Hídricos*, 12(3).
- Haykin, S., 2009. *Neural Networks and Learning Machines*. Prentice Hall.
- Lachtermacher, G. e Fuller, J. D., 1994. Backpropagation in hydrological time series forecasting. In: *Stochastic and Statistical Methods in Hydrology and Environmental Engineering*, v. 10/3, p. 229–242. Springer Netherlands.
- Maceira, M. E. P., Penna, D. D. J., Vasconcelos, C. L. V., Ennes, M. I. A., e Lima, A. L. D. S., 2013a. *Determinação da Coordenação da Operação a Curto Prazo - DECOMP, Manual do Usuário*. Centro de Pesquisas de Energia Elétrica, Rio de Janeiro.
- Maceira, M. E. P., Penna, D. D. J., Vasconcelos, C. L. V., Ennes, M. I. A., e Lima, A. L. D. S., 2013b. *Modelo Estratégico de Geração Hidrotérmica a Subsistemas Interligados - NEWAVE, Manual do Usuário*. Centro de Pesquisas de Energia Elétrica, Rio de Janeiro.
- Rumelhart, D. E., Hinton, G. E., e Williams, R. J., 1986. Learning representations by back-propagating errors. *Nature*, v. 323:p. 533–536.
- Sacchi, R., 2009. *Política de Operação Preditiva Estabilizada via Termo Inercial utilizando "Analytic Signal", "Dynamic Modelling" e Sistemas Inteligentes na Previsão de Vazões Afluentes em Sistemas Hidrotérmicos de Potência*. Tese de Doutorado, Univesidade de São Paulo - Escola de Engenharia de São Carlos.
- Silva, E. L., 2001. *Formação de Preços em Mercados de Energia Elétrica*. Sagra Luzzato.
- Silva, I. N., Flauzino, R. A., e Spatti, D. H., 2010. *Redes Neurais Artificiais para engenharia e ciências aplicadas*. Artliber Editora Ltda.